

Wire-speed Multi-Dimensional Packet Classifier

by inventors

Rodolfo Milito, Adolfo Nemirovsky, and Mario Nemirovsky

5

Field of the Invention

10 The present invention is in the field of digital processing and Internet routing devices, and pertains more particularly to apparatus and methods for packet classification and processing.

0
9
8
7
6
5
4
3
2
1
0
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0
20

Background of the Invention

25 The present invention is in the area of routing devices in packet networks, such as the well-known Internet network, and in the tasks of identifying and processing packets in routing devices. Packets in such a network are logical groupings of data. A packet includes a header, typically having several fields which contain identity and control information. A separate part of the packet contains the main information to be transmitted by a path determined in the routing process.

25

Routing actions taken by network routing devices are governed by pre-programmed rules, and a typical network routing device can have a large number of rules. A network routing device makes routing decisions based on information coded in header fields of a packet to select a rule that applies to that packet. As described above, a typical network routing device can have many rules and can exercise a number of actions on a packet including, but not limited to routing, dropping, queuing and labeling.

The header fields in a packet may have as many as 128 bits for the next generation Internet Protocol, known in the art as Internet Protocol version 6 (IPv6). Ipv6 is a proposed replacement protocol for the current version of Internet Protocol, referred in the art as IPv4, or Internet Protocol version 4.

The mapping of the values of certain header fields to the set of rules is referred to in the art as packet classification, which is the technology area of the present invention. The function of packet classification enables network managers to specify policies that identify network traffic in order to partition, or classify that traffic into multiple priority levels, and is at the core of functionality in the new generation of network routing devices.

Packet classification is an active area of research, the speed and efficiency of which has been determined by the inventors to have limitations bound by current technology. Previous work in this field has provided some solution to the problem, one example being the Lakshman and Stiliadis solution, summarized below in more detail. This solution is described in a paper entitled "High-Speed, Policy Based Forwarding Using Efficient Multi-dimensional Range Matching" published by Proc. ACM SIGCOMM September 1998, pp 191-202. This paper is incorporated herein by reference.

Packet classification is a conceptually relatively simple problem turned difficult by the combined demands of speed, dimensionality and size of the fields in the current and impending technology. In terms of speed there is a growing need for processing packets at wire speeds in ports operating at OC-48 and higher. In terms of dimensionality the number of rules may be in the range of thousands, and even hundreds of thousands. The number of fields in IPv4 to be examined for classification is up to 5, and each field can be up to 32 bits long. When newer versions of Internet protocol are used

(IPv6), there may well be more fields of greater length and the problems will be multiplied.

What is clearly needed is a method and apparatus for faster and more efficient multi-dimensional mapping of header fields in a packet to a rule or set of rules, and that overcomes the speed limitations in packet classification that exist in current technology. The present invention builds upon previous work in the art, providing a solution to the long-standing problem of the limitations inherent in current technology, at a cost of very little added logic in a system design.

5

10
15
20
25

Summary of the Invention

In a preferred embodiment of the present invention a system for classifying packets, wherein each packet has N header fields to be used for processing is provided, the system comprising a first set of rules associating to the packets by values of the header fields; and a classification system for selecting specific rules in the set of rules as applicable to a specific packet. The system is characterized in that the classification system projects the first set of rules as N -dimensional entities on N axes in N -dimensional space, marking the beginning and ending value on each axis for each rule as a breakpoint, numbers intervals between breakpoints in sequential binary numbers, associates a subset of the first set of rules as applicable in each interval between breakpoints on each axis, then considers a packet as a point in the N -dimensional space according to its header field values, locates the binary numbered interval into which the point projects on each axis by performing a search on each axis for the numbered interval into which the point projects on that axis, thereby determining rules applicable to the packet

20

25

for that axis, and then determines the specific rules applicable to the packet from the subsets of rules by selecting those rules as applicable to the packet that apply to the packet on all of the N axes.

In one preferred embodiment the search performed on each axis is a binary search conducted by selecting breakpoints at which the bits change for the binary numbered intervals. In other embodiments the search performed on each axis is a quaternary or higher-level M-ary search, where M is a power of 2, conducted by selecting breakpoints at which the bits change for the binary numbered intervals.

In some embodiments association of applicable rules in each numbered interval is made by associating a binary string with each interval, with one bit dedicated to each rule. Also in some embodiments the rules are associated to bit positions in the binary string by priority, the order of priority according to bit significance, and a final rule is selected by the most significant 1 in the matching rules. In preferred embodiments the applicable rules are found by ANDing the binary strings determined for each axis over all axes.

In some embodiments there is at least one hardware pipeline for conducting the search on an axis, the pipeline comprising first, second, and sequential modules for accomplishing increasingly particular portions of the search, wherein, after the first module of the sequential modules is used, determined values from the first module pass to the second module, and values for a second packet enter the pipeline at the first module, the pipeline operations proceeding thus sequentially. There may also parallel pipelines with one pipeline dedicated to searching on each axis in the N-dimensional space, wherein searches are conducted for applicable intervals simultaneously on each axis. Also, applicable rules for each interval on each axis may be represented by individual bitmaps, with each rule assigned a bit

10
09
08
07
06
05
04
03
02
01
00

5

20

25

position, and the outputs of the parallel pipelines, being the numbered interval on each axis into which the point for a packet projects, may be exchanged for the associated bitmaps, which are then ANDed to determine the applicable rules.

5 In some embodiments of the invention searching is interleaved, such that results of searching on one or more axes being applied to other axes before searching on the other axes. In some interleaving embodiments rules that are found by search to not apply on one or more axes are not considered in searches conducted on the other axes.

In another aspect of the invention a method for classifying packets in routing, wherein each packet has N fields to be used in processing in a header is provided, comprising the steps of (a) projecting the rules as N-dimensional entities on N axes in N-dimensional space; (b) marking the beginning and ending value on each axis for each rule as a breakpoint; (c) numbering intervals on each axis sequentially with binary numbers; (d) identifying those breakpoints at which bits in the interval numbers change; (e) associating a subset of the rules as applicable in each interval on each axis; (f) considering a packet as a point in the N-dimensional space according to values of the header fields for the packet; (g) determining by search the binary numbered interval on each axis into which the packet point projects; (h) substituting the subset of rules that apply for each determined interval; and (i) selecting those rules as applicable to the packet that associate to the packet on all of the N axes.

20 In some embodiments of the invention, in step (g), the determination is made by a binary search. Also in some embodiments, in step (g), the determination is made by a quaternary or higher-level M-ary search. In some embodiments of the method, in step (e), association of applicable rules in each numbered interval is made by associating a binary string with

10
15
20
25
30
35
40
45
50
55
60
65
70
75
80
85
90
95

20

25

each interval, with one bit dedicated to each rule. The rules may be mapped to bit positions in the binary string by priority, the order of priority according to bit significance, and a final rule is selected by the most significant 1 in the matching rules. The matching rules are found by ANDing the binary strings determined for each axis over all axes in step (i).

In one embodiment of the method, in step (g), the search is conducted by sequential modules in at least one hardware pipeline, the pipeline comprising first, second, and sequential modules for accomplishing increasingly particular portions of the search, and, after the first module of the sequential modules is used, determined values from the first module pass to the second module, and values for a second packet enter the pipeline at the first module, the pipeline operations proceeding thus sequentially. In this embodiment there may be parallel pipelines with one pipeline dedicated to searching on each axis in the N-dimensional space, with searches conducted for applicable interval simultaneously on each axis.

In some embodiments applicable rules for each interval on each axis are represented by individual bitmaps, with each rule assigned a bit position, and the outputs of the parallel pipeline, being the numbered interval on each axis into which the point for a packet projects, are exchanged for the associated bitmaps, which are then ANDed to determine the second set of matching rules. In some embodiments, in step (g), searching is interleaved, results of searching on one or more axes being applied to other axes before searching on the other axes. In these embodiments rules that are found by search to not apply on one or more axes may not be considered in searches conducted on the other axes.

In another aspect of the invention, in a system for classifying packets by binary or higher-level searching for intervals into which rules project on axes, a method for simplifying a search is provided, comprising the steps of

(a) conducting a first search on one or more axes; and (b) using information from the first search to simplify further searching on remaining axes.

In various embodiments of the present invention taught in enabling detail below, for the first time a very fast and reliable method and apparatus is provided for mapping rules to packets in a packet routing device.

5

Brief Description of the Drawings

10

Fig. 1 is a mapping of three rules onto two axes representing two header fields for a packet.

15

Fig. 2 is a table relating breakpoints in the mapping of Fig. 1 with interval numbers and bitmaps of rule association by interval.

20

Fig. 3 is the graphical representation of Fig. 1 with a specific packet represented

Fig. 4 is a table illustrating a first search step in relating rules to a packet.

25

Fig. 5 is a table representing a second step in relating rules to packets.

30

Fig. 6 is a table representing a third step in relating rules to packets.

Fig. 7 is an illustration of a pipelined process for processing packets in an embodiment of the present invention.

35

Fig. 8a is an illustration of branching in a binary search process.

40

Fig. 8b is an illustration of branching in a quaternary search process.

45

Fig. 9 is an illustration of an alternative pipelined process in an embodiment of the present invention.

Description of the Preferred Embodiments

5

In the Lakshman and Stiliadis solution referred to above there are two phases: a pre-processing phase and a packet-by-packet phase. In the pre-processing phase rules are considered as multi-dimensional entities, there being as many dimensions as there are header fields to be used in classification in packets to be processed, and the rule dimensions are projected onto Cartesian axes.

10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
289
290
291
292
293
294
295
296
297
298
299
299
300
301
302
303
304
305
306
307
308
309
309
310
311
312
313
314
315
316
317
318
319
319
320
321
322
323
324
325
326
327
328
329
329
330
331
332
333
334
335
336
337
338
339
339
340
341
342
343
344
345
346
347
348
349
349
350
351
352
353
354
355
356
357
358
359
359
360
361
362
363
364
365
366
367
368
369
369
370
371
372
373
374
375
376
377
378
379
379
380
381
382
383
384
385
386
387
388
389
389
390
391
392
393
394
395
396
397
398
399
399
400
401
402
403
404
405
406
407
408
409
409
410
411
412
413
414
415
416
417
418
419
419
420
421
422
423
424
425
426
427
428
429
429
430
431
432
433
434
435
436
437
438
439
439
440
441
442
443
444
445
446
447
448
449
449
450
451
452
453
454
455
456
457
458
459
459
460
461
462
463
464
465
466
467
468
469
469
470
471
472
473
474
475
476
477
478
479
479
480
481
482
483
484
485
486
487
488
489
489
490
491
492
493
494
495
496
497
498
499
499
500
501
502
503
504
505
506
507
508
509
509
510
511
512
513
514
515
516
517
518
519
519
520
521
522
523
524
525
526
527
528
529
529
530
531
532
533
534
535
536
537
538
539
539
540
541
542
543
544
545
546
547
548
549
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
988
989
989
990
991
992
993
994
995
996
997
998
999
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2079
2080

For illustrative purposes, more than three dimensions are difficult to represent graphically, and fields with a large number of bits are cumbersome as well, but the principles of the Lakshman and Stiliadis method as well as features of the present invention may be described for practical purposes in two dimensions. In the following example there are two five-bit header fields for packets, and three rules.

20

Fig. 1 is a Cartesian representation of three rules labeled (1), (2) and (3) projected on an X-axis and a Y-axis each having a hexadecimal range of 0 to 1F, there being 5 bits in each header. The X-axis represents one of the two header fields, in this case labeled Field (A), and the Y-axis represents the other field, in this case labeled field (B). The position and order of the axes is arbitrary. This representation is reasonable, as rules must associate to header values.

25

In this rather simple example the upper and lower field value boundaries for each rule are projected onto each axis, creating a series of *breakpoints* on each axis, the breakpoints establishing a series of *intervals* on each axis. Consider rule (2) for example. Rule (2) is known to potentially

apply to a packet if the value of Field (A) for a specific packet falls between 0E and 1C (HEX), that is, between binary 01110 and binary 11100. Rule (2) projects on the Y-axis for field (B) in the interval between 04 and 09 (HEX). The projections of rules (1) and (2) are similarly shown on the axes.

5 To avoid confusion in these examples, the rules are considered to include the breakpoints. That is, if a header value falls on a breakpoint projected by a rule, that rule is considered to apply. Other conditions may apply in other cases.

10 In this illustration rule (3) is contiguous (and all rules are contiguous); that is, rules (1) and (3) overlap. In this example there are seven intervals cast on each axis, including the maximum dimension 1F as a breakpoint. For N rules, the maximum number of intervals on an axis, including the maximum dimension as a breakpoint, will be $2N+1$, or in this particular example, 7.

20 Now, also in the pre-processing phase, an N-dimensional bitmap is created and associated with each interval. This bitmap, in this case of 3 bits (N=3), denotes which rules apply relative to the specific interval on the particular axis. For example, in the interval 03 to 07 on the X-axis for field (A) in Fig. 1, both rules (1) and (3) are associated, but not rule (2). The bit map for interval 03 to 07 is therefore 101. A 1 in the ith position indicates that rule i is associated with that interval. The bit order relative to rules is arbitrary, and our example relates bits left to right for fields in ascending numerical order. That is, a 1 in the first bit place from the left (most significant bit) indicates rule 1 is associated in the particular interval.

25 Fig. 2 is a table created for the intervals on the two axes in our example. There are three columns in the table. The leftmost column shows interval breakpoints, which are the *endpoints* for each interval (compare with X-axis and Y-axis of Fig. 1) The physical interval in the table is that interval

with the listed breakpoint as its endpoint, and the previous breakpoint as its start point.

The middle column in the table of Fig. 2 is a binary number in ascending order from 0 for each interval on each axis. Note for example that for breakpoint (endpoint) 01 for the X-axis, for which the interval is 0 to 01, the interval number is 001. The intervals are numbered to provide, in a preferred embodiment of the invention, a unique way for structuring the process of determining into which interval a header value for a packet in process falls.

The rightmost column in the table of Fig. 2 is the bitmap for the interval, which relates the rules that apply for that interval. In the case of interval number 001, from 0 to breakpoint 01, the bitmap is 000, as no rule projects on the X-axis in this interval (see again Fig. 1).

The skilled artisan will be able to follow the breakpoints, interval ordering, and bitmaps for the rest of the X-axis and for the Y-axis for the table of Fig. 2 in this example.

It needs to be said at this point that the pre-processing phase, including all projections, interval ordering, and bitmaps, remains stable as long as the rule set is stable, and needs to be edited and updated only when rules change. In some cases the rules will change frequently, and in others the rules will change only at longer intervals. In a routing device the rules may change for any of a number of reasons, such as load factor, time-of-day, and so on. There may be software for monitoring conditions and changing the rule set that applies, or rule changes may be accomplished by manual input.

In the packet classification process, given a specific set of rules, and assuming the pre-processing phase is done, resulting in the table of Fig. 2, operation proceeds in the packet-by-packet phase. In the packet-by-packet phase, packets are taken one-at-a-time, and the applicable rule(s) are

determined according to the values of the header fields (two fields considered in this example).

Fig. 3 is the same as Fig. 1, except a packet in process is represented by a point X in the two-dimensional space. The point is located by the field values for fields A and B. It may be assumed that this packet has been acquired by the system for the purpose of determining the rule which is to be used to process the packet. Although two fields, thus two dimensions, are used in this example, the skilled artisan will recognize that a packet may be represented by a point in N-dimensional space, such as in as many as five dimensions for IPv4.

In Fig. 3 the packet acquired for processing has a field value of 10 (HEX) in Field (B) and 05 (HEX) in Field (A). To determine the applicable rule or rules, binary searches are done in a preferred embodiment of the invention, typically in parallel for the two axes (fields). The object of the binary searching is to determine the interval on each axis within which a projection of the point X falls. Considering, for example, the X-axis, this is done by selecting pertinent break points (being the projections of edges of rules on the axis), and determining, step-by-step, whether the projection of the packet point is greater than or less than the break point. By process of elimination the interval into which the point projects can be isolated incrementally.

A unique contribution in a preferred embodiment of the present invention is in determining the best break points and methods to accomplish the search in the least number of steps. There are, of course, a number of ways one may select among the breakpoints and the search may be conducted, some of which are less reasonable than others.

As an example of a relatively inefficient method, one might select among the existing breakpoints without preference, and do a compare of the

5

10
15
20
25
30
35
40
45
50
55
60
65
70
75
80
85
90
95

20

25

5 selected breakpoint value with the packet point projection, yielding where the point lies relative to the selected breakpoint. Referring again to Fig. 3, consider, for example, a first step on the X-axis using the breakpoint 1C. A compare will show that the point 05 lies to the left of 1C, eliminating the interval from 1C to 1F. One may then select any one of the breakpoints between 01 and 1C, and continue the process. Eventually the correct interval will be isolated.

10 Another possibility is to select breakpoints considering the binary value of the breakpoints, at a point at where the most significant bit of the X-value changes. In the present example, 0E is 01110 and 17 is 10111. In this scheme one would select 17 as the first breakpoint. In this scheme the search continues by selecting breakpoints on the axes where the second bit changes, the third bit changes, and so on to the fifth bit.

20 The present inventors, however, have determined an improved process, and have elected to number the intervals sequentially in binary, and to select breakpoints by the sequentially numbered intervals (middle column in Fig. 2). This scheme has an advantage in that there are three bits in the sequential interval numbers (in this example) rather than five bits to deal with in the axis values. In a preferred embodiment of the present invention the steps in the search proceed as follows:

25 Step 1: Breakpoint 07 is selected on the X-axis precisely because the interval number (middle column in Fig. 2) of all intervals on the X-axis to the left of (less than) this breakpoint have a 0 in the most significant bit (MSB) of the interval number, and the interval number of all intervals to the right of (greater than) this breakpoint have 1 in the MSB. Breakpoint 09 is selected for the first step for the Y-axis because the interval number of all intervals above 09 on the Y-axis have 1 as the MSB and the interval number of all

intervals below 09 on the Y-axis have 0 as the MSB. Step 1 in the binary search for each field axis is represented in Fig. 4, with the result, which is a pointer to the next step. This operation for the X-axis compares the value of field (A), which is 05 for the packet in process, to the breakpoint 07. Since 05<07 it is determined that the MSB of the interval number of the interval in which the field value lies is 0. A similar comparison on the Y-axis, using the breakpoint 09, at which value the MSB changes, compares 10 to 09, and yields 1, because 10>09. These values from step 1 become pointers to step 2 for each axis. The step for the X-axis and the Y-axis are done in parallel in a preferred embodiment, and for all axes in cases with many more axes.

After Step 1 the MSB of the interval number into which the point projects on each axis is determined.

Step 2: The table of Fig. 5 illustrates step 2. In the first column are the possible values of the pointer from step 1 (either 1 or 0 in this example) for each field. The second column is for the breakpoint to determine the next MSB, which is the middle bit of the three-bit interval number sought in this example. Referring to Fig. 2 and Fig. 3, it is seen that, for the X-axis, if the pointer to step 2 is 1, the value of field (A) on the x-axis will be in an interval between 07 and 1F, while if the pointer to the second step is 0 the value of Field (A) will be in an interval between 00 and 07. In the range from 07 to 1F, the value of the second bit for the interval numbers changes at 17, so 17 is the selected breakpoint. In the interval between 00 and 07 the breakpoint for the middle bit is 01. For the Y-axis for field (B), if the pointer to the second step is 1 the interval sought is an interval in the range from 09 to 1F, and the selected breakpoint is 13, where the second MSB changes value. If the pointer to the second step for the Y-axis is 0, the value lies in the overall interval from 00 to 09, and the breakpoint for the middle bit is 03. So step 2

compares the value of Field (A) with 17 if the pointer is 1 and to 03 if the pointer is 0. If the pointer is 1 and the field value is > 17 , the pointer to the third step is 11. If the field value is < 17 the pointer to the third step is 10. If the pointer is 0, and the field value is > 01 , the pointer to the third step is 01, and if the < 01 the pointer to the third step is 00. Similarly, for the Y-axis for field (B), if the pointer is 1 the comparison is the value of field (B) with 13. If the value is > 13 the pointer to the third step is 11, and if the value is < 13 the pointer to the third module is 10. If the pointer to the second step is 0, the comparison is with 03, and if $>$ the pointer to the third step is 01, and if $<$ the pointer to the third step is 00.

For the packet in process in this example, having field (A) = 05 and field (B) = 10, the pointer to the third step is 01 for field (A), because $05 > 01$; while the pointer to step 3 for field (B) is 10, because $10 < 13$. The first and second MSBs for the interval numbers sought on each axis are now determined.

Step 3: Fig. 6 illustrates step 3 in general. The first column is the entry point, being the pointer from the second step. The second column is the breakpoint at which the value of the least significant bit (LSB) of the interval number changes for the overall interval in which the value is known to lie, and the comparisons and results are shown in the third column. In the present example the pointer from the second step for field (A) is 01, and for field (B) is 10. For field (A) $05 > 03$, so the final result of the binary search on the X-axis is interval number 011. For field (B) $10 < 0B$, so the final result for the Y-axis is interval number 101.

Step 4: Step 4 relates the rules to the packet in process by virtue of the interval numbers on the two axes in which the point determined by the field

5

10

SEARCHES
00000000000000000000000000000000

20

25

values for the packet project. This is done by entering the table (Fig. 2) which relates numbered intervals to the bitmaps that associate rules to intervals. The binary searches on the axes have determined the numbered interval on each axis within which the point determined by the field values of the packet in process lie. Entering the table of Fig. 2 it is seen that the binary bit map relating rules to intervals has bit map 101 for interval 011 on the Y-axis and bit map 101 for interval 101 on the Y-axis.

5

10
9
8
7
6
5
4
3
2
1

Step5: The fifth step combines the bit map for the interval on the X-axis within which the point projects, with the bit map for the interval on the Y-axis within which the point projects. This a logical **AND** operation, which yields 101 **AND** 101 = 101.

The final result for this rather simple example is the bit map 101, indicating that rules 1 and 3 both potentially apply to the packet for which the field values of the header are 05 and 10 for field (A) and field (B) respectively.

20

Since two rules potentially apply but just a single rule must be selected in this example, there is default logic to select the applicable rule. In this example, when more than one rule applies, the rule of MSB applies. The MSB of the **AND** result (101) is for rule 1, so rule 1 is applied to the packet in process. In other embodiments there may be other defaults and algorithms for tie-breaking when more than one rule potentially applies. Also, there will be a default for the result wherein no rule is found to apply in the classification process.

25

It will be apparent to the skilled artisan that the illustration would be considerably more complex for as many as five fields of up to 32 bits each and a large number of rules (IPv4), and even more complex for developing

Internet protocols for future use. The example provided, however, fairly illustrates and teaches the method in a preferred embodiment of the invention.

In the embodiment of the present invention described in step-by-step progress above, as each packet arrives to be processed in the packet-by-packet phase, the several steps are performed and the best rule is selected for that packet, then another packet is processed. Again, as before, if the rules change, the mapping of the rules to axes has to change as well (pre-processing phase) before further packets may be processed. Of course, tables for multiple rule sets may be stored, and the correct table selected when rules change.

In another embodiment of the invention a significant improvement is made in the packet-by-packet phase. This improvement results from the present inventors discovering that the step-by-step parallel process is amenable to a pipelined structure and operation.

Fig. 7 is a structure and flow diagram for a pipelined search implementation in an embodiment of the present invention. There are three modules in each pipeline, labeled modules 1, 2, and 3, and two pipelines, one for each axis in our simple example. In other cases, depending on the number of rules and header fields, the number of modules in a pipeline and the number of pipelines may also change. The modules in a preferred embodiment are cascaded hardware structures with associated registers for changing breakpoints and other data.

Firstly, in the pre-processing phase, rules are projected on the axes, and intervals are determined and numbered. The appropriate breakpoints for MSB, middle bit and LSB are determined and stored, and the first breakpoints (MSB) for module 1 for each dimension (axes X and Y) are loaded into modules 1 for each pipeline. Referring back to the step-by-step

5

SEARCH 15002000

20

25

process taught above, it will be clear to the skilled artisan that the module 1
breakpoint will not change. The breakpoints used for succeeding modules
will depend on the result of the immediately preceding modules. There are
several ways this may be handled. In some embodiments each module has
5 hardware structure for each possibility from the preceding module. In other
embodiments the result (pointer) from a preceding module selects the
breakpoint for the next module as processing proceeds.

As an example of the structure and operation of sequential modules,
reference is made again to the steps described above with respect to Figs. 2-
5. The first breakpoint for the X axis is 07 and the first breakpoint for the Y
axis is 09 (Fig. 4). X and Y from Field (A) and Field (B) for a first packet
are fed into Module 1 of each pipeline. Module 1 for the X-axis determines
on which side of 07 the point falls, and module 1 for the Y-axis determines
on which side of 09 the point falls. Module 1 for each pipeline generates a
pointer to the second module, and passes the point values X and Y of the
first packet to the second module. Depending on the result of the compare
in the first module, the correct breakpoint is set for the compare to be made
in the each of the second modules. At the same time header field values for
a second packet are loaded into module 1 for each axis.

20 In some embodiments the hardware structure allows for all possible
breakpoints, which are loaded into the pipeline modules in the pre-
processing phase. There are, in this case alternate paths in the hardware for
the second module, and the path is selected by the value of the pointer from
module 1 for each pipeline. In this example the alternative breakpoints for
25 module 2, which are 17 or 01 for the X axis, and 13 or 03 for the Y-axis (see
Fig. 5). The correct path is taken based on the pointer from module 1 in
each pipeline. Module 2 for each axis determines the middle bit for the

10
15
20
25
30
35
40
45
50
55
60
65
70
75
80
85
90
95

interval sought for the packet-in-process at module 2. At the same time module 1 for each axis is determining the MSB for a new packet.

Module 2 now passes a pointer and the X and Y values to module 3 for each axis. At the same time module 1 passes a pointer and the X and Y values for the second packet-in-process to module 2, and values for a third packet are loaded into module 1 for each axis.

Module 3 may allow for alternative hardware paths for all of the possibilities from module 2 for each axis, or the pointer values may be used to select the correct breakpoints to be loaded to the third modules in each pipeline (see Fig. 6).

Module 3 for each axis determines the LSB for the first packet-in-process. The interval number on each axis is now known for the first packet, as is shown at the output of module 3 for each axis in Fig. 7.

In a next cycle, knowing the interval number, a table lookup returns the rule-association bitmap for the interval number determined for each axis for the first packet, and at the same time new values are loaded into the three modules as described above (see Fig. 2). In each cycle, a step is taken for each packet in each pipeline. At the end of the two pipelines an AND operation resolves the bitmaps (in this case 2 bitmaps) into one bitmap, and the correct rule is selected by default logic. In the present example the logic is that the rule associated from the AND operation with the MSB is the rule to be applied to the packet.

The pipelined operation proceeds, loading a new point (header field values for a new packet to be processed) into modules 1, and moving point values and pointers to next modules, and determining the rule to be used for packets emerging from the pipeline, as long as the rules do not change. At a rule change new breakpoints are determined as appropriate, which also

5

10

0123456789FS0123456789

20

25

proceeds in a sequential fashion across the pipeline, assuring that the right
breakpoints are used for the right packets in process.

There are advantages (throughput) in many cases to accomplishing as
much as possible with hardware and pipelined structure. As the structure is
hardware, however, the structure itself may not be readily changed physically
in a particular machine. The number of header fields for packets, however,
remains constant over long periods of time. As long as determinations are
being made for IPv4 packets, for example, the number of header fields to be
used in classification is up to 5 and the field length is up to 32 bits.

Therefore the number of modules provided will be, in a preferred
embodiment, enough to accommodate the situations expected to be
encountered. The inventors believe, at the time of the present filing, that ten
modules will be adequate for most embodiments of the invention. For
applications where fewer modules are needed, there will be provision for
taking the output of the last needed module and feeding that into the table
lookup for rule association, leaving some pipeline modules idle. The skilled
artisan will recognize there are a number of ways this may be done.

In alternative embodiments of the invention the pre-processing phase
for rules changes may be done in any of a variety of ways. For example,
rules changes may often be incremental rather than drastic. There will
typically be known rule sets with which to deal as well. In preferred
embodiments known rule sets will be stored, together with pre-selected
breakpoints and other data associated with or calculated from the rule sets,
according to fields and headers for packets to be processed, and provision is
made for very rapid allocation of breakpoints, and so forth, at the times that
pre-processing is needed. It will also be true that there may be cases where
the rules change, but the change will not effect, or will not seriously effect

5

10

20

25

the application of rules to packets, and there is no need to recalculate or redistribute breakpoints for the pipeline structure.

In another aspect of the present invention the inventors have determined that the search process, which is amenable to pipelining, is also amenable to an M-ary search, where M is a power of 2. One might do a quatenary search, for example, and in a specific application a quatenary (or higher-level) search may be advantageous. In following description a quatenary search will be used as an example, but the inventors intend that the description can also apply to higher-level searches as well. In binary searches one bit is determined at a time. In a quatenary search two bits, and in an M-ary search, where $M=2^k$, and $k=1, 2, \dots, n$, the search determines k bits at a time.

Fig. 8a and 8B illustrate the decision paths in a binary search and a quatenary search, respectively. In the binary search shown in Fig. 8a, from start the decision path is either a or b, then c or d if a, or m or n if b, and so on. At each decision point the path goes either one way or the other of two possibilities. In the quatenary search of fig. 8b there are four alternative paths at each decision point.

In general for the quatenary search the logic for a hardwired module is more robust. Also, there needs to be more than one break point considered. The decision in the quatenary search typically involves logic of the sort: IF $a > b$ AND $c > d$, then e, OR if $a < b$ AND $f > g$, then h, OR (and so forth). The logic can be worked out and implemented in silicon to do the quatenary search, and the necessary structures are within the ability of those with ordinary skill in the art.

The inventors have provided in another embodiment of the present invention yet another novel way to do a search, for those cases when the

10
095364175
060200
0020

20

25

circumstances warrant, and it can be done with structure little different in hardware than that designed for binary searching.

Fig. 9 is a pipeline structure similar to that illustrated in Fig. 7 having three modules and two pipelines, which is capable of resolving intervals for which the structure of Fig. 7 would require six modules. In Fig. 9 the indication of registers for breakpoints as illustrated in Fig. 7 has been removed to avoid the drawing becoming confused, but the breakpoint registers are still associated with the modules. In the pipelines of Fig. 9 the basic hardware structure is essentially the same as in Fig. 7, and the operations are much the same, except each module is used twice. The return loop arrow shown from the output of each module back to the input of the same module illustrates this repeated use.

The pipelines of Fig. 9 operate as follows: Field values for a first packet enter module 1, and module 1 now has access to the first breakpoint for the MSB. the module outputs a pointer indicating the compare for the MSB, and that pointer is fed back into module 1 as indicated by the return loop arrow. At the same time the pointer is fed back the breakpoint is indexed to the breakpoint to find the next most significant bit. The operation of the same module then determines the next significant bit for the interval number. After the second pass for a module, the pointer goes to the next module.

The net effect of the pipelines of Fig. 9 is that each module provides a double step and the overall pipeline length is shortened. It is possible, to use a single binary search module any number of times, rather than twice as described above, and many alternative structures are provided for different situations, saving silicon real estate and gates at the expense of latency in indexing breakpoint values and the like. There are situations and

5

40
39
38
37
36
35
34
33
32
31
30
29
28
27
26
25
24
23
22
21
20

20

25

circumstances where this may be advantageous, and circumstances where the longer pipelines may be advantageous.

In yet another aspect of the present invention, the inventors have determined that there may be special circumstances wherein interleaving between pipelines may be in order. Consider, for example, the case where a large number of rules may be disqualified in a search done on one axis. Since, to be applicable to a packet, a rule has to be applicable on *all* axes, any rules that do not apply on one axis do not have to be considered on another axis. If the pipeline process is performed entirely in parallel, then the search is done for all rules in the parallel pipeline for each axis, yet the search might be greatly simplified (fewer intervals for projection of fewer rules) if the pre-processing were redone for a second axis after eliminating rules in a search done on the first axis. There would, of course, be a penalty of the loss of the advantage of the parallel pipelining. In some cases the reduction in time for succeeding searches might more than offset the penalty of the loss of parallelism.

There are a great variety of ways that interleaving might be done. For example, in one embodiment, all of the rules may be projected on a first axis, breakpoints determined, intervals numbered, and then a single step-by-step process using an appropriate number of modules is used to complete a search on the first axis for a candidate packet. The pipeline may be constructed in any of the ways herein discussed; for instance to perform a binary search, a quaternary search, to reuse modules, and so forth.

In this first exemplary embodiment, once the interval is determined on one axis in which the first packet projects, the table lookup is done for the bit map that associates rules to intervals, and the bitmap is saved. Now use is made of the information just determined, that the candidate packet associates with certain rules as a result of its projection on the first axis, but

5

10
15
20
25
30
35
40
45
50
55
60
65
70
75
80
85
90
95

equally importantly, there is an entire set of rules with which the packet does not associate. Returning to Fig. 3 and the associated descriptions above, it may be seen that rule 2 is ruled out in the first step. The candidate packet having X(05) and Y(10) projects on the X-axis in consecutively numbered 5 interval number 011, and rule 2 does not apply.

The search on the Y-axis may now be simplified. Only two rules are still candidates after the search on the X-axis, so the number of intervals on the Y-axis is fewer (5 instead of 7). The search on the Y-axis, then, will require fewer modules, and the rule association on the Y-axis may be determined more quickly than on the X-axis. Once the bit map is determined for the Y-axis, it is ANDed with the saved bit map from the X-axis, and the final rule selection is made.

It will be apparent that in more sophisticated situations, wherein there are many rules, there may be situations where a first pass on a single axis will eliminate most of the rules. In such situations perhaps only very few additional axes may have to be considered until only one rule (or no rule) is found to be applicable, at which point the classification is complete.

In still another alternative embodiment, after a first pre-processing phase, in which rule projections are made, intervals are numbered, and breakpoints assigned to whatever hardware structure is provided for the search function, short test searches are made to determine the apparent advantage of various approaches. For example, one makes a short search on the X-axis utilizing just one binary search module, and records which (and how many) rules are eliminated. The same short search is then done for a second axis, and the result compared with the result of the first short search. When (and if) a first short search yields a large sacrifice of rules, that is, a great proportion of the rules are ruled out, then the intervals are recast on the remaining axes, and the searches continued. In this embodiment, as a 20 25

10
11
12
13
14
15
16
17
18
19

further refinement, once a large number of rules are ruled out, the pre-processing phase is redone, and the search proceeds with the full complement of parallel pipelines., requiring a significantly foreshortened search process.

5

The inventor notes here, that in these embodiments and variations of these embodiments, once a first pre-processing phase is done for the full contingent of rules, it is not necessary that the pre-processing be redone because certain rules are eliminated as candidates for a packet in process. Rather, the tables for numbering, rule association, and the like can be altered in a systematic manner, because all of the information required on any axis for a subset of the original rules will be in the information for the full set of rules.

10
0
9
8
7
6
5
4
3
2
1

20

In yet another slightly different embodiment, a set of breakpoints may be defined based simply on the range on the axes (a function of the number of bits in a header field for a packet), rather than by projecting the rules on the axes. These defined (and constant for range) breakpoints may simply divide the axis into equal-length intervals of any convenient number, preferably in powers of two. For example, 16 intervals. Foreshortened searches may be made on the basis of these defined intervals to determine expected advantage, then the projected interval breakpoints may be used in the subsequent long search.

25

In still other embodiments, there will be statistical operations and other historical functions. In these embodiments separate logic determines load factors and trends for types of packets, and applies selectivity in classification operations based on statistical variations. For example, if the statistical operations determine that a great preponderance of packets are of the same type, source, and destination over a period of time, then the

classification process may be greatly simplified until the mix and load factors change.

The skilled artisan will realize that there are a large number of alterations that might be made in the embodiments described herein, and that different designers might design the hardware and procedures differently in many cases, while staying well within the bounds of spirit and scope of the present invention. The scope of the invention, then, should be limited only by the claims which follow.